

# XDDS:

## A Scalable Guard-Agnostic Cross Domain Discovery Service

**Michael Atighetchi**, Raytheon BBN Technologies

**Joseph Loyall**, Raytheon BBN Technologies

**Jonathan Webb**, Raytheon BBN Technologies

**Michael J. Mayhew**, AFRL/RIEB

**Abstract:** As both the DoD and the Intelligence Community (IC) are moving toward service-oriented architecture (SOA), it is important to ensure that SOA-based systems can operate and exchange classified information across domain boundaries in support of net-centric missions. The interplay between SOA and cross domain solutions (CDS) raises a number of challenges that are grounded in the inherent mismatch between core SOA principles, such as loose coupling, composability, and discoverability, and current CDS technologies and certification and accreditation processes in use today. The Cross Domain Discovery Service (XDDS) described in this paper provides an architecture and design for extending service discovery, a core SOA functionality, across domain boundaries. The resulting services and protocols provide access to service information across security domains in a secure, guard-agnostic, scalable, and flexible way that is amenable to certification and accreditation (C&A).

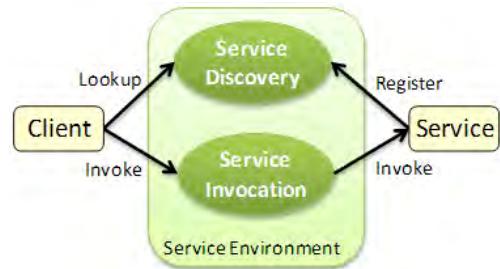
### Introduction

SOA is becoming increasingly important to, and entrenched in, the DoD and IC for military and intelligence operations, including initiatives such as Net-Centric Enterprise Services (NCES). While SOA includes services and support for security, such as access control, these initiatives have largely concentrated only on providing security within domains,<sup>1</sup> not across them. Simultaneously, CDS<sup>2</sup> have begun to handle the growing requirement to service the need to share information critical to military operations, disaster response, national intelligence, and other situations, carefully balancing the need to share with the traditional need to protect sensitive or classified information within and across domains.

Discovery services play an important role in single domain SOAs because of the dynamic nature of a service environment. As services become available, change, or get removed, applications need to have up-to-date information about the definition of available services. Management of static depictions of these environments becomes difficult, both within and across domains, particularly as the number of services increases. This motivates a requirement for discovery services across domains that is currently unmet by existing service discovery solutions, which only work within domains.

Discovery itself is a simple process, as shown in Fig. 1. A service registers itself with the service discovery service that is part of an existing service environment. Next, a client (shown on the left) performs lookup requests on the service discovery service to find newly registered services. Once the client has found a suitable service, it proceeds to invoke that service through a specific invocation mechanism.

*Fig. 1. Functional View of the Discovery Process within a Single Domain*



The XDDS described in this paper fills this gap by enabling dynamic discovery and use of services across a variety of domains and associated relationships, including hierarchical, non-hierarchical, and coalition. The resulting services and protocols provide access to service information across security domains in a secure, guard-agnostic, scalable, and flexible way that is amenable to C&A following standard IC and DoD processes, e.g., DIACAP [1], NIST Special Publication 800-53 [2], or ICD 503 [3]. The XDDS prototype addresses requirements expected of any new cross domain capability in an early research and development prototype lifecycle. The XDDS prototype is:

- >> Guard-agnostic, i.e., independent of any specific guard implementation.
- >> Modular, enabling reuse of existing guards and services that have successfully passed C&A
- >> Developed, documented, and tested with C&A in mind and an explicit goal to provide a body of evidence for certification and accreditation processes in later phases.

### DoD-Centric Use Case

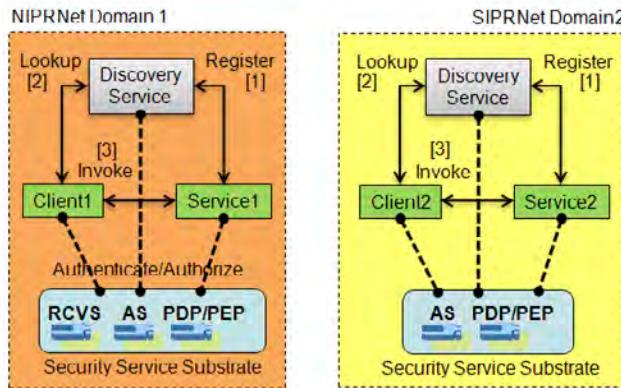
Current support for net-centric operations is based on isolated deployments of relevant services in individual domains. Fig. 2 illustrates a NIPRNet and SIPRNet deployment of the DISA NCES service discovery service, which provides the ability to register services (in step 1), lookup services

Report Documentation Page			Form Approved OMB No. 0704-0188					
<p>Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p>								
1. REPORT DATE <b>NOV 2010</b>	2. REPORT TYPE	3. DATES COVERED <b>00-00-2010 to 00-00-2010</b>						
<b>4. TITLE AND SUBTITLE</b> <b>XDDS: A Scalable Guard-Agnostic Cross Domain Discovery Service</b>			5a. CONTRACT NUMBER					
			5b. GRANT NUMBER					
			5c. PROGRAM ELEMENT NUMBER					
<b>6. AUTHOR(S)</b>			5d. PROJECT NUMBER					
			5e. TASK NUMBER					
			5f. WORK UNIT NUMBER					
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> <b>Raytheon BBN Technologies,Information and Knowledge Technologies Unit,Cambridge,MA,02138</b>			8. PERFORMING ORGANIZATION REPORT NUMBER					
<b>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b>			10. SPONSOR/MONITOR'S ACRONYM(S)					
			11. SPONSOR/MONITOR'S REPORT NUMBER(S)					
<b>12. DISTRIBUTION/AVAILABILITY STATEMENT</b> <b>Approved for public release; distribution unlimited</b>								
<b>13. SUPPLEMENTARY NOTES</b>								
<b>14. ABSTRACT</b> <p>As both the DoD and the Intelligence Community (IC) are moving toward service-oriented architecture (SOA), it is important to ensure that SOA-based systems can operate and exchange classified information across domain boundaries in support of net-centric missions. The interplay between SOA and cross domain solutions (CDS) raises a number of challenges that are grounded in the inherent mismatch between core SOA principles, such as loose coupling, composability, and discoverability, and current CDS technologies and certification and accreditation processes in use today. The Cross Domain Discovery Service (XDDS) described in this paper provides an architecture and design for extending service discovery, a core SOA functionality, across domain boundaries. The resulting services and protocols provide access to service information across security domains in a secure, guard-agnostic, scalable, and flexible way that is amenable to certification and accreditation (C&amp;A).</p>								
<b>15. SUBJECT TERMS</b>								
<b>16. SECURITY CLASSIFICATION OF:</b> <table border="1"> <tr> <td>a. REPORT <b>unclassified</b></td> <td>b. ABSTRACT <b>unclassified</b></td> <td>c. THIS PAGE <b>unclassified</b></td> </tr> </table>			a. REPORT <b>unclassified</b>	b. ABSTRACT <b>unclassified</b>	c. THIS PAGE <b>unclassified</b>	<b>17. LIMITATION OF ABSTRACT</b> <b>Same as Report (SAR)</b>	<b>18. NUMBER OF PAGES</b> <b>8</b>	<b>19a. NAME OF RESPONSIBLE PERSON</b>
a. REPORT <b>unclassified</b>	b. ABSTRACT <b>unclassified</b>	c. THIS PAGE <b>unclassified</b>						

(step 2), and finally invoke services (step 3), but only within a respective domain. Extending the discovery of services across domains necessitates introduction of new cross flows for either disseminating service registrations or lookup requests. The flows also need to contain filters to ensure that clients only get access to information they are entitled to, even from remote domains.

As part of the XDDS effort described in this paper, we designed and prototyped services and cross domain protocols that enable Client1 in Fig. 2 to discover and use Service2, and conversely Client2 to discover and use Service1, if and only if these interactions are permissible under existing cross domain data sharing policies.

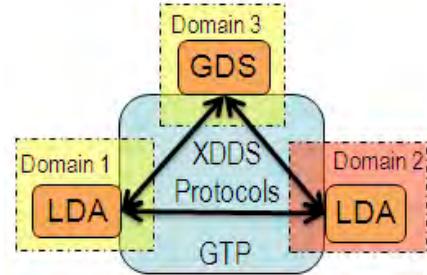
*Fig. 2. Current Service Discovery in DoD Enterprise Environments*



## XDDS Architecture

We created an extensible and flexible architecture for cross domain service discovery and implemented versions of the following components shown in Fig. 3.

*Fig. 3. XDDS Architecture*



A Local Discovery Agent (LDA) in each domain transparently intercepts lookup and registration requests within domains and handles routing requests across domains for performing local discovery, e.g., using Universal Description Discovery and Integration (UDDI). The LDAs enable transparent discovery across domain boundaries without requiring code changes to existing discovery services or clients.

The XDDS message protocol, through which LDAs interact with one another, is based on two simple generalized communication models, namely referral- and replication-based

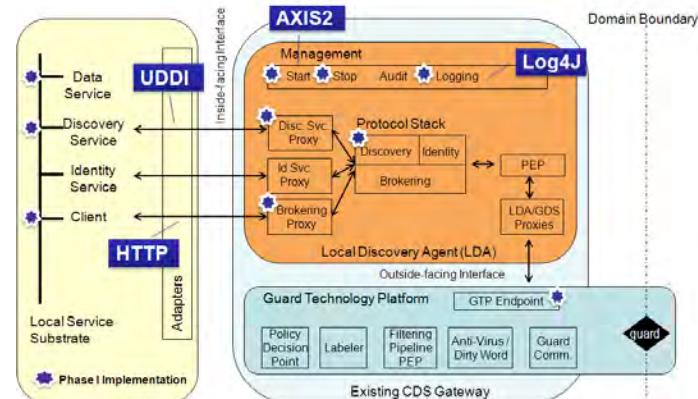
discovery, and accommodates requirements on message exchanges in cross domain environments, such as restricted XML schemas. Our protocol can encapsulate a large number of variant discovery protocols without significant changes, minimizing the impact of changes on C&A.

The Global Discovery Service (GDS) is an extended LDA component that facilitates scalability by introducing hierarchy through which any number of LDAs can interact. The GDS maintains information about the domains and how they can reach one another, enforces policies on cross domain interactions, facilitates proper authentication, and supports anonymization of domains.

The Guard Technology Platform (GTP) is a generic interface to existing guards supporting the examination of cross domain flows represented with the XDDS protocol. Using the GTP abstraction during development allows XDDS to remain independent of specific guard and CDS implementations.

Fig. 4 shows the XDDS architecture in more detail. The LDA is strategically placed between the local service substrate on the left, e.g., an ESB, and the guard, which is located closest to the cross domain boundary on the right. Interfaces of the LDA to other components can be categorized into inside-facing and outside-facing. While the inside-facing interfaces talk to existing services in the local domain through adapters, the outside-facing interface interacts with the local endpoint of the Guard Technology Platform within an existing CDS gateway such as the Collaboration Gateway [4] or the Web Service Gateway [5] of the Cross Domain Collaborative Information Environment [6].

*Fig. 4. The LDA and its Interfaces to Local Service Substrates and CDS Gateways*



For communicating with the local service substrate, the LDA instantiates service proxies and uses adapters for supporting a number of different protocols. In Phase I, we implemented proxies for the discovery and brokering and implemented adapters for UDDI v2 and HTTP.

For communication with other LDAs through the outside-facing interface, the LDA uses LDA service proxies. Since the Phase I prototype only involved two LDAs, we directly linked the LDA with the GTP endpoint through the Simple Object

# IMPORTANT STANDARDS AND GUIDANCE DOCUMENTS

- NIST Special Publication 800-95, Guide to Secure Web Services, 8/2007
- MITRE Technical Report MTR080027, Recommendations on the Use of SOAP in a Cross Domain Environment, 2/2008
- MITRE Technical Report MTR040000092, Security Guards for the Future Web, 9/2004
- NSA Report XML Schema Guidance for Cross Domain Security Policy Enforcement, 07/2006
- Intelligence Community Standard for Information Security Marking Metadata (IC ISM), ICS 2007-500-2, Version 2, April 2004
- SOAP 1.1 (W3C Note 08 May 2000) and SOAP 1.2 (W3C Recommendation 27 April 2007)
- WSDL 1.1 Specification (W3C TR Note 14 March 2001)
- OASIS UDDI Technical Note Using WSDL in a UDDI Registry, Version 2.0.2
- OASIS UDDI V2 specification

Access Protocol (SOAP). The LDA process itself is implemented as a web service and hosted in the Axis2 [7] web services container. Logging is performed using the Log4J [8] framework.

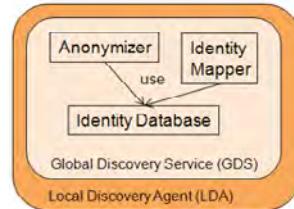
The heart of the LDA is its layered protocol stack, with a brokering protocol at the lowest layer and discovery and identity management at higher layers. The main purpose of this stack is to convert messages from the local service substrate, which may be complex and technology specific, into a core set of simple messages, which are suited for crossing domain boundaries. One way to describe the collection of core messages is via the notion of abstract protocols for discovery, identity, and brokering. The brokering protocol is responsible for routing requests through the network of XDDS nodes. Requests can either be discovery or identity management related, or originate from a brokering proxy which allows clients to invoke services in other XDDS-enabled domains. Discovery is implemented via exchange of simple messages that allow for registration of local services with XDDS and lookup of registered services throughout the overlay network of XDDS nodes. The identity protocol enables an LDA to export a selected set of identity mappings from the local identity service to other XDDS nodes (such as the GDS).

The LDA contains a Policy Enforcement Point (shown as PEP in Fig. 4) that intercepts requests and subjects them to policy evaluation. In future versions of the LDA, we expect to configure an existing Policy Decision Point with role-based access control policies that determine what information is allowed to be passed outbound and received inbound. To meet the requirement for preventing data leakage between domains, the policy enforcement of high domains always happens on the high side, allowing domains to stay in full control of their data. In addition, low domains implement a second line

of defense by pushing protection requirements closer to the source of misbehavior in cases of errors or attacks that are mounted to escalate from low to high domains.

The GDS is built on the same technology platform as the LDA to provide support for anonymization, identity mapping, and LDA synchronization (as displayed in Fig. 5). The discovery service in a GDS operates at a higher layer in the discovery hierarchy in that it manages LDA memberships and allows them to discover each other. LDAs defer to the GDS for discovery requests that they cannot handle and answer discovery queries from the GDS. For identity management, the GDS supports mapping of identities across domains in a scalable way. It can also store identity relevant meta-information about LDAs, such as what identity protocols are supported by an LDA and whether the LDA allows remote verification of identities. For anonymization, the GDS supports multiple operational modes, ranging from traditional onion-routing to support for services that want to disclose only a small subset of information about themselves and implement “don’t call us, we’ll call you” policies. The GDS allows XDDS to support service discovery even in the most restrictive environments in which the knowledge that a certain domain hosts a certain service is not permitted to cross domain boundaries.

*Fig. 5. The GDS*



## Cross Domain Service Discovery In Action

To ensure feasibility of the XDDS architecture and design and construct a body of evidence for later C&A activities, we implemented a proof-of-concept prototype during Phase I based on the jUDDI open-source server [9].

We started by constructing a baseline scenario for intra-domain discovery of Web Services Description Language (WSDL)-described web services following the WSDL in UDDI OASIS recommendation [10]. We then proceeded to implement referral-based discovery across two domains. Key components of the prototype include an implementation of the XDDS protocol specification together with a set of configurable transformations on UDDI and WSDL documents necessary for cross domain discovery.

The set of transformations, implemented using XSLT, includes scripts to change service end point information, e.g., for making cross domain service calls via existing cross domain web service invocation substrates, as well as to restrict information sharing due to security restrictions, e.g., by redacting UDDI operator identities. The prototype allows flexible control over content and location of transformations applied to the message stream and also rejects messages that do not conform to expectations, e.g., by analyzing sequence numbers to prevent replay attacks.

Fig. 6. Proof-of-Concept Prototype Demonstration

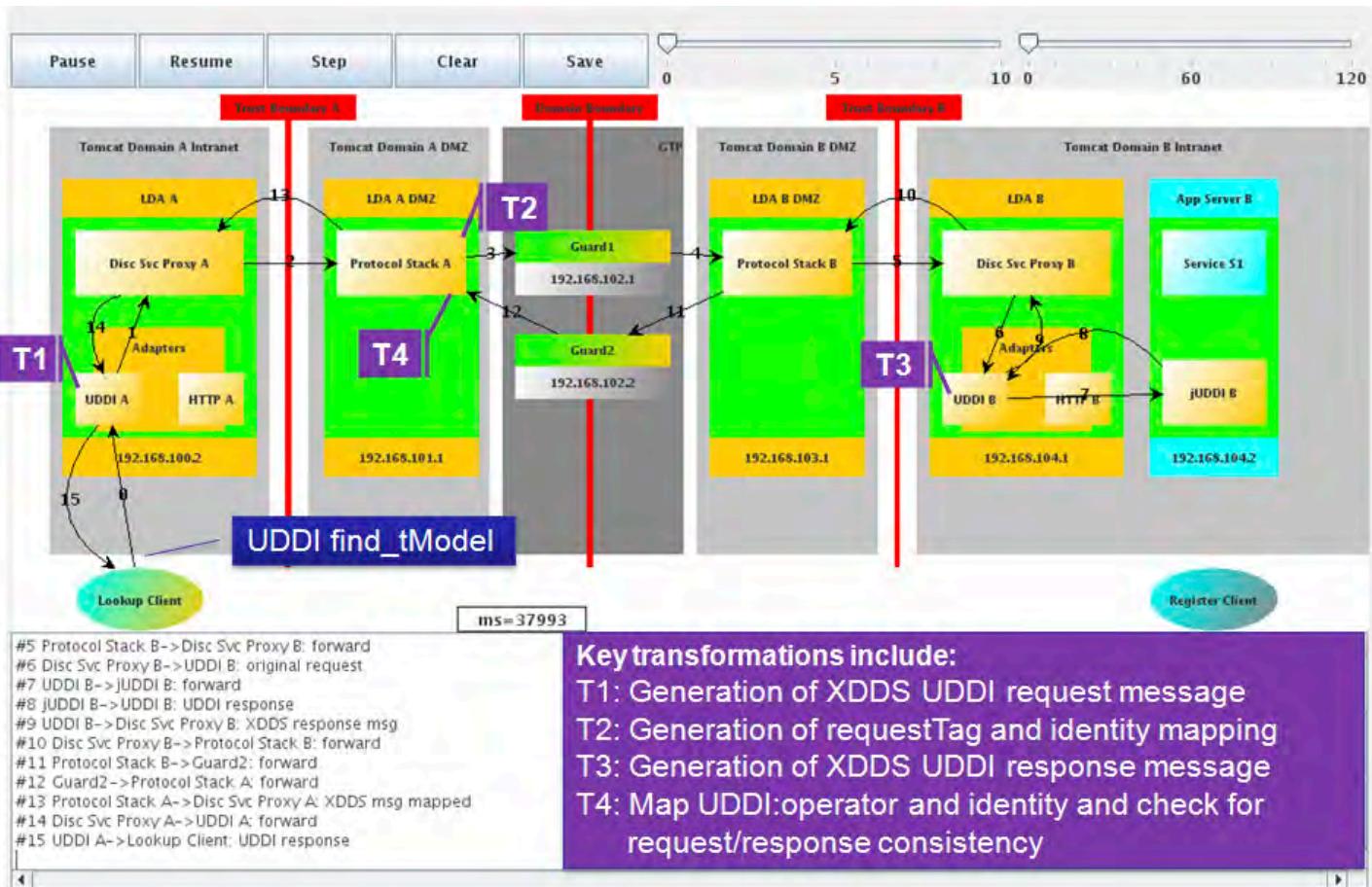


Fig. 6 shows a visualization of the multi-step cross domain discovery process generated from live outputs and logs of participating components. The domain boundary is shown in the center and the GTP is represented through a dark gray box. LDA components are further divided into an intra-net resident LDA process, e.g., LDA A, and a process resident in a Demilitarized Zone<sup>3</sup> (DMZ), e.g., LDA A DMZ. The lookup client is represented by an oval on the left, while the UDDI server is represented by an orange box labeled "jUDDI B" on the right. Fig. 6 shows the sequence of XML message exchanges between various components during a UDDI find\_tModel<sup>4</sup> together with key transformations on the resulting XDDS messages called out via T1 through T4.

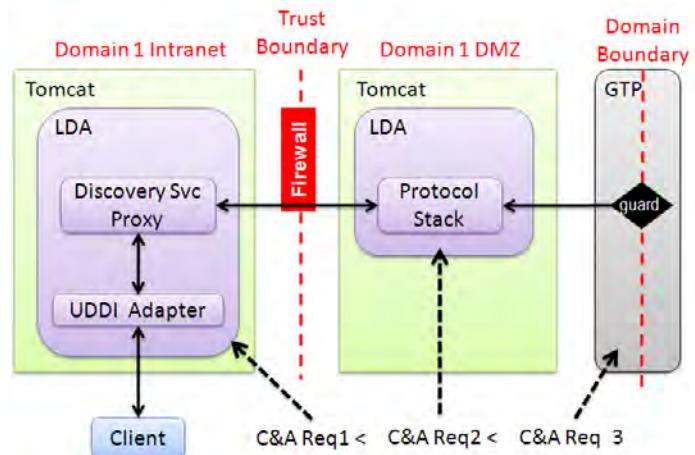
### Certification and Accreditation of Different Configurations

C&A of CDSs is of significant cost and solutions that do not account for the specifics of cross domain environments will face significant barriers during accreditation. This is even more true for service discovery due to a high degree of technology diversity and proliferation of evolving discovery standards. To address these issues, XDDS decouples existing discovery technologies found locally in a domain from the messages that cross the domain boundary.

The design of the Phase I prototype confines most of the complexity to the protocol adapter, specialized for UDDI in this

case, and the discovery service proxy while allowing the LDA/GDS components to exchange a small set of core XDDS XML messages within a narrowly defined message format over the domain boundary (the right side of Fig. 7). Message exchanges across domain boundaries are represented via two generalized communication models, referral and replication (described in more detail later in this section), that cover a wide variety of discovery protocols through adapters.

Fig. 7. Separation of LDA Components along Trust Boundaries





The scope of C&A in this effort was to construct an initial body of evidence that can be used later as the basis for security arguments for a real C&A activity. The various design tradeoffs, use cases, and XML message exchanges and transformations shown through the proof-of-concept prototype all feed into construction of this body of evidence. In addition, we developed the design and proof-of-concept prototype to be consistent with a number of important community documents and standards.

### Functional Use Cases and Generalized Communication Models

XDDS supports two basic discovery patterns: referral, where a client request is transferred from a local proxy to a discovery service instance holding the relevant service registration, and replication, where service registrations are copied to local discovery service instances to satisfy local discovery requests.

#### Basic Discovery Interaction Patterns

The Phase I proof-of-concept prototype supports referral-based discovery, in which the LDA components disseminate lookup requests and corresponding responses across domain boundaries, as shown in Fig. 8.

The sequence of events is as follows:

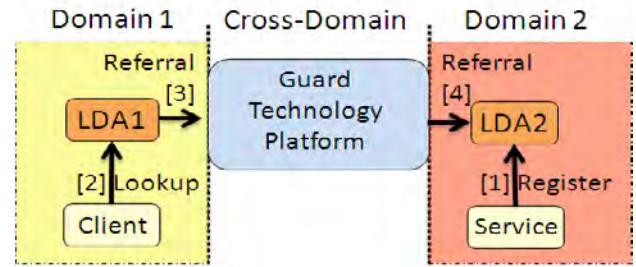
- 1) A service in Domain 2 makes a registration request with its local LDA 2. Service description information is only persisted locally.
- 2) A client in Domain 1 makes a lookup request with its local LDA 1.
- 3, 4) The LDA 1 forwards the request to LDA 2 in the other domain and receives the response back from LDA 2, which it in turn returns to the client. The transfer of cross domain requests and response is mediated by the GTP.

Fig. 9 depicts the replication-based discovery configuration, and the sequence of steps is as follows:

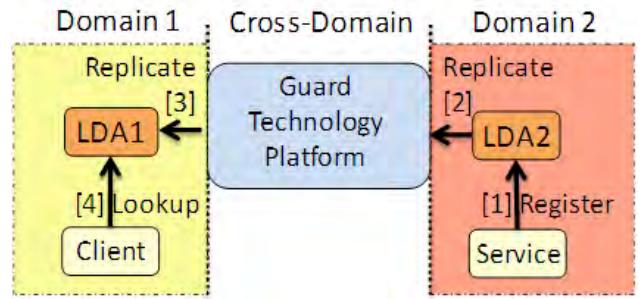
- 1) A service in Domain 2 makes a registration request with its local LDA 2.
- 2, 3) The LDA 2 makes a replication request through the GTP to an affiliated LDA in another domain. Transfer is mediated by the GTP.
- 4) The replication request is received by the affiliated LDA 1 and any local client requests are serviced by the LDA 1 local to the client C.

Note that a GDS (in its own domain) may be inserted into the communication path to reduce or eliminate the need for multiple point-to-point connections. XDDS provides mixed

*Fig. 8. Referral-based Discovery*



*Fig. 9. Replication-based Discovery*



operations in which one LDA is configured to replicate registrations to the GDS, while another LDA uses referrals for lookup operations. In both configurations, XDDS carries all cross domain message exchanges over a generic discovery service protocol.

There are two key distinctions between the referral and replication models:

For referral, the information traverses the domain boundary at the time the lookup request is made by the recipient client. For replication, the information traverses the domain boundary at the time the service registration is performed.

In the referral model, the service description is not in the persistent storage of the discovery service element of the requesting domain. The replication model has a persistent copy of the discovery data in all replication domains.

The differences have important implications on security aspects of deployments. For example, it may be more appropriate to replicate service registrations from low to high domains. In this configuration, lookup requests performed in the high domain are handled locally, reducing the amount of risk for interference or covert channels.

## XDDS Protocol Specification

The XDDS message protocol is an XML-based message specification that describes the syntax of messages passed between LDAs through the GTP. The protocol is consistent with open standards, e.g., XML, UDDI, Security Assertion Markup Language, WS-Security, XML Signature, and SOAP. The protocol represents XML message exchanges through two basic message forms—XDDS requests (example shown in Fig. 10) and XDDS responses. By default, all requests generate responses and generic acknowledgement responses are returned in error cases instead of error responses. Messages include control information, such as LDA identities used for routing purposes, classification markings, and message integrity and provenance trails that allow enforcement of integrity and anti-spoofing.

*Fig. 10. Example XDDS message*

```
<xdds:xdds xmlns="http://xdomain.bbn.com/xdds/"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:uddi="urn:uddi-org:api_v2"
  xmlns:wsdl="http://schemas.xml.org/wsdl/" xmlns:xdds="http://xdomain.bbn.com/xdds"/>
<classificationMarking classification="U"/>
<requestLabel>
  <domainIdentity>DomainA</domainIdentity>
  <requestTag>req_0</requestTag>
</requestLabel>
<responderIdentity>
  <domainIdentity>DomainB</domainIdentity>
</responderIdentity>
<discoveryQuery>
  <uddiV2Query>
    <find_tModel generic="2.0" xmlns="urn:uddi-org:api_v2">
      <name xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
        <categoryBag xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
          <keyedReference keyName="WSDLtype" keyValue="binding_tModelKey="uuid:6e090afa-33a5-36eb-81b7-1ca18373f457">
            <keyedReference keyName="binding namespace" keyValue="http://quickstart.samples/" tModelKey="uuid:d01987d1-ab2e-3013-8be2-2a66eb99d824"/>
          </categoryBag>
        <find_tModel>
          <uddiV2Query>
            <discoveryQuery>
              <xdds:xdds>
```

To simplify messaging formats, the protocol uses the same message types during referral and replication modes and treats the replication request analogous to a query response in the referral mode. Furthermore, application specific discovery protocols, e.g., UDDI and HTTP, are encapsulated in the XDDS messages in restricted form, allowing the same XDDS message structure to be used with multiple application specific protocols.

The XDDS protocol allows expression of restrictions on message exchanges through both XML schema and XSLT restrictions. The schema restrictions include sanitization of the input stream through removal of non-printing characters and any characters outside the range 040 to 176. In addition,

we use white space normalization and disallow any CDATA, Base64, or other similar binary encodings. The protocol handlers further restrict attribute values to enumeration constants or highly constrained value sets and disallow mixed element content. Extensible Stylesheet Language Transformations (XSLT) restrictions are generated automatically from configuration data and tie allowable message exchanges to accredited cross domain flows. For instance, the XDDS protocol handlers use XSLT script to check message ordering and detect message replay scenarios.

## Summary and Next Steps

The XDDS Phase I project was a successful research effort that produced significant improvements in technology in a short amount of time. The technology innovations and the XDDS prototype demonstrated in this project are foundational results enabling a necessary capability, previously unavailable, if SOA is to be realizable in DoD and Intelligence Community environments, namely the ability to discover and broker services across domain boundaries in a scalable, safe, and certifiable manner.

In summary, we designed a guard-agnostic architecture for cross domain service discovery based on the principles of modularity, interoperability, transparency, scalability, and security, and produced technical designs for its major components, namely the LDA, the GDS, and the XDDS protocol specification. In addition, we developed use cases involving generalized communication models, namely referral-based and replication-based discovery, advanced discovery capabilities, including hierarchical and anonymous discovery processes, and assured discovery capabilities through authentication and authorization, message protection through signatures, and traffic restrictions and normalization. Finally, we successfully demonstrated cross domain discovery via a proof-of-concept implementation of one specific configuration supported by the design.

Our plans for future work include expansion of the existing proof-of-concept capabilities by a) adding replication-based discovery and initial authentication, authorization, and service invocation capabilities, b) developing the first version of the GDS component to enable hierarchical discovery and enhance the replication and referral-based discovery capabilities to support message integrity and pedigree, and c) implementing anonymization and providing enhanced management and generation of variant configurations.♦



**Michael Atighetchi** is a scientist at BBN's Information and Knowledge Technologies business unit. His research interests include cross domain information sharing, security and survivability architectures, and middleware technologies. Mr. Atighetchi has published more than 35 technical papers in peer-reviewed journals and conferences, and is a senior member of the IEEE. He holds a master's degree in computer science from the University of Massachusetts at Amherst, and a master's degree in IT from the University of Stuttgart, Germany.

**Raytheon BBN Technologies**  
**10 Moulton Street**  
**Cambridge, MA 02138**  
**Phone: (617) 873-1679**  
**Fax: (617) 873-4328**  
**E-mail: matighet@bbn.com**



**Dr. Joseph Loyall** is a Principal Scientist at Raytheon BBN Technologies. He has been the Principal Investigator on DARPA and USAF AFRL R&D projects in the areas of information management, distributed middleware, adaptive applications, and quality of service. He is the author of over 75 published papers; was the program committee co-chair for the Distributed Objects and Applications conference (2002, 2005); and has been an invited speaker at several conferences and workshops. He has a Ph.D. from the University of Illinois.

**Raytheon BBN Technologies**  
**10 Moulton Street**  
**Cambridge, MA 02138**  
**Phone: (617) 873-4679**  
**Fax: (617) 873-4328**  
**E-mail: jloyall@bbn.com**



**Jonathan Webb** is an engineer in BBN's Information and Knowledge Technologies business unit. Over 20 years at BBN, Mr. Webb has been involved in a wide range of software development projects including simulation of dynamic systems, web-based data management systems, middleware for information management, and cross domain information sharing. Mr. Webb has a master's degree in aeronautics and astronautics from the Massachusetts Institute of Technology.

**Raytheon BBN Technologies**  
**10 Moulton Street**  
**Cambridge, MA 02138**  
**Phone: (617) 873-3321**  
**Fax: (617) 873-4328**  
**E-mail: jwebb@bbn.com**



**Michael J. Mayhew** has been with AFRL for the past 13 years, 6 of those years as a Federal Civilian. As the Program Manager of the Cross-Domain Innovation & Science group, Michael leads a team of research engineers in finding and developing new cross domain technologies and maturing those technologies for integration within existing cross domain products. Michael is a frequent presenter at worldwide program and technical conferences each year and is recognized as a subject matter expert in the area of cross domain technology. In addition to his CDIS role, Michael also serves as the Science & Technology Liaison between his group and the DoDIIS Cross Domain Management Office. Michael's long career included Government Lead Engineer on the ISSE Guard 3.5 project and Senior Computer Analyst with Northrop Grumman Government Systems. Michael received an M.S. with Magna Cum Laude in Computer Science from SUNY IT. Michael is certified ACQ Level 1 Certified Program Manager and ACQ Level 3 Certified SPRD&E. Michael is a member of the Cross Domain Solutions Working Group, and the Armed Forces Communications and Electronics Association, Erie Canal Chapter.

**AFRL/RIEB**  
**525 Brooks Road**  
**Rome, NY 13441-4505**  
**Group Tel: (315) 330-7380**  
**Tel: (315) 330-2898 DSN: 587-2898**  
**Fax: (315) 330-3913 DSN: 587-3913**  
**E-Mail: michael.mayhew@rl.af.mil**

## ACKNOWLEDGMENTS

The authors would like to acknowledge the support and collaboration of the U.S. Air Force Research Laboratory (AFRL) Information Directorate. This material is based upon work supported by the AFRL under Contract No. FA8750-09-C-0012.

## NOTES

1. A Domain represents one or more computers under the same specific security policy.
2. A Cross Domain Solution is an approved trusted data flow implemented between two or more domains.
3. A DMZ is a physical or logical subnetwork that contains and exposes an organization's external services to a larger untrusted network, e.g., the Internet.
4. The find\_TModel UDDI request is used to retrieve summary information about UDDI tModel elements describing a service.

## REFERENCES

- DoD. (2007) Signed DoDI 8510.01 - Department of Defense Information Assurance Certification and Accreditation Process (DIACAP) Instruction. <<http://www.dtic.mil/whs/directives/corres/pdf/851001p.pdf>>.
- NIST. (2009, August) Special Publication 800-53. <<http://csrc.nist.gov/publications/PubsSPs.html>>.
- Director of National Intelligence (DNI) Directive, Intelligence Community Directive (ICD) 503, 2008.
- Boyd Fletcher, USJFCOM J9/SPAWAR, "XMPP & Cross Domain Collaborative Information Environment (CDCIE)," in *Overview For Net-Ready Sensors Summer Workshop Collaboration Gateway*.
- Chris Roberts, Kurt Risser, Boyd Fletcher, "The Design and Implementation of a Guard Installation and Administration Framework," in SE-Linux Symposium, 2007.
- JFCOM. (2009, Nov.) Cross Domain Collaborative Information Environment. [Online]. <[http://www.jfc.com.mil/about/fact\\_cdcie.html](http://www.jfc.com.mil/about/fact_cdcie.html)>.
- Apache Software Foundation. (2010, January) Apache2 Homepage. [Online]. <<http://ws.apache.org/axis2/>>.
- Apache Software Foundation. (2010, January) Log4j Homepage. [Online]. <<http://logging.apache.org/log4j/1.2/index.htm>>.
- Alan Vinh and Phil Bonderud. (2008, Dec.) jUDDI. [Online]. <<http://ws.apache.org/juddi/>>.
- OASIS UDDI Spec TC. (2004, June) Using WSDL in a UDDI Registry - Version 2.0.2 - Technical Note. [Online]. <<http://www.oasis-open.org/committees/uddi-spec/doc/tn/uddi-spec-tc-tn-wsdl-v2.htm>>.



# Homeland Security

# Software Assurance

**Software is essential to enabling the nation's critical infrastructure.**

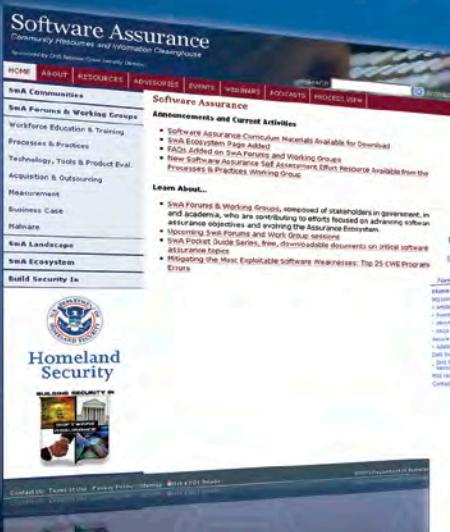
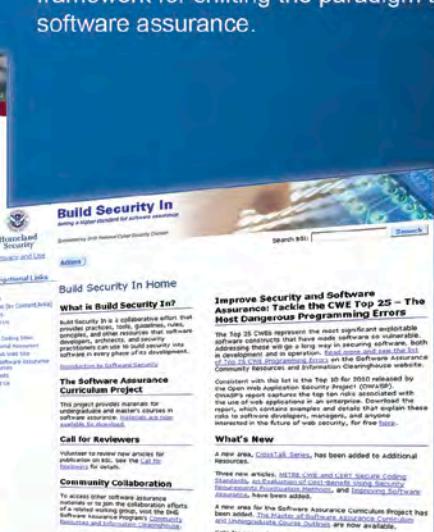
To ensure the integrity of that infrastructure, the software that controls and operates it must be secure and resilient.

Software Assurance Community Resources and Information Clearinghouse provides corroboratively developed resources. Learn more about relevant programs and how you can become involved.

**Security must be “built-in” and supported throughout the lifecycle.**

Visit <https://buildsecurityin.us-cert.gov> to learn about the practices for developing and delivering software to provide the requisite assurance. Sign up to become a free subscriber and receive notices of updates.

The Department of Homeland Security provides the public-private collaboration framework for shifting the paradigm to software assurance.

<https://buildsecurityin.us-cert.gov/swa>